第六章-色调映射

图像显示技术的最终目的就是使得显示的图像效果尽量接近人们在自然界中观察到的对应的场景。HDR 图像与视频有着更高的亮度、更深的位深、更广的色域，因此它无法在常见的普通显示器上显示。入门级的显示器与播放设备（例如普通人家使用的电视，常见的电脑、智能手机屏幕等）的对比度很低，只有大约 200：1。相对性能更好的 LCD 显示器能达到更高的对比度，大约 10000：1。但是，这些设备通常都会将一个色彩通道离散化到 8-bit，少数 10-bit 的色度区间内。这意味着色度区间只有 255 个层级，这样的显示设备，我们称之为 LDR 显示设备。很显然，HDR 图像是无法在这些显示设备上直接显示的，为了使得 HDR 图像与视频也能兼容地放映在 LDR 设备上，就必须使用色调映射（Tone Mapping）技术将 HDR 图像与视频映射为 LDR 图像与视频，进而可以正常显示在 LDR 显示设备上。

尽管近年来 HDR 显示设备的研究与生产已经逐步进入正轨，并且部分厂商生产的 HDR 电视或者屏幕已经投入市场，但是不可忽视的是，如今大多数显示设备仍然属于 LDR 显示设备，HDR 显示设备想要实现对 LDR 显示设备的全面取代，仍然需要经历一个漫长的过程。因此，色调映射技术在这样的背景下，就显得尤为重要。

色调映射的目的是使高动态范围 HDR 图像能够适应低动态范围 LDR 显示器。想要实现这一目的，就需要将 HDR 图像的色度，亮度，动态范围等，全部映射到 LDR 图像的标准范围内。在这个过程中，不能简单使用线性映射，因为这样会丢失原始图像的一些重要信息，例如全局与局部的对比度和重要的细节等，最重要的一点是，简单线性映射产生的图像相比于自然界中的场景，会出现严重的失真情况。因此，色调映射技术需要经过深入的研究与精密的设计，才能使得产生的图像在视觉效果上与实际场景中的效果一致。本章的前半部分，我们将首先对色调映射算法的整体思想与相关内容作出综述，接着通过图像色调映射算法，介绍清楚色调映射算法的基本思想与设计方法，以及常见或常用的图像色调映射算法。

随着图像色调映射算法的发展日益成熟，另一个更加复杂但是却同样具有实际意义的问题也随之出现，那就是视频色调映射算法。视频色调映射算法本质上可以看作对视频中的每一帧应用图像色调映射算法，但是，由于视频独有的时域特性以及时域联系，简单地将图像色调映射算法应用于视频当中是不合理的，并且也会使得视频质量出现很大的问题，例如帧间闪烁现象。因此，视频色调映射算法与图像色调映射算法虽然存在联系，但是也有所区别。在本章的后半部分，我们会介绍色调映射算法在视频方面的独特运用。

6.1 色调映射算法综述

色调映射算法的目的在于将 HDR 图像的亮度进行压缩，进而映射到 LDR 显示设备的显示范围之内，同时，在映射的过程中要尽量保持原 HDR 图像的细节与颜色等重要信息。所以色调映射算法需要具有两方面的性质：1. 能够将图像亮度进行压缩。2. 能够保持图像细节与颜色。

对于 HDR 图像与 LDR 图像来说，两者的亮度是存在区别的，这种区别不仅仅在于范围上，也在于亮度的意义上。对于 HDR 图像来说，亮度值就代表绝对的亮度值。由于 HDR 显示设备的亮度范围足够大，因此，HDR 图像每一个像素点记录的亮度值，就

是显示在 HDR 显示设备上的亮度值。以 OpenEXR 格式的 HDR 图像为例：OpenEXR 格式的图像的亮度范围在6.14×10^{-5}~6.41×10^{4}。这些亮度信息记录在一个单独的亮度信道中，显示在 HDR 显示器上的亮度，就是亮度信道中记录的亮度。而 LDR 图像则有所不同，因为 LDR 显示设备的亮度范围很小，并且编码格式也是 0-255 的整数，因此能够显示的颜色与亮度的范围都很有限。为了让有限的亮度空间都能被合理利用，LDR 图像的亮度实际上并不对应于显示场景中的亮度，而是一种“相对亮度”。像素值通过 Gamma 映射，映射到显示器支持的亮度范围当中后，再显示出来。因此，LDR 图像的亮度是由显示设备决定的，而 HDR 图像的亮度记录在图像当中，由图像决定的。

由于 LDR 显示设备的亮度范围远远小于 HDR 图像以及自然界中的亮度范围。因此，在设计色调映射算法时，如果仅仅使用简单的线性映射的方法，将会产生的问题是过于明亮与过于阴暗区域的细节无法保持，这显然是不可接受的。所以，在设计算法时，必须考虑图像中场景的显示一个场景时，主要需要考虑其光照水平和对比度范围，一些需要普遍关注的例子有：在阳光充足的场景下，事物会显得更加多彩和对比强烈；而夜晚时，多彩的事物会显得灰暗的多。因此，对像素的简单缩放和压缩其光照强度水平和对比度范围不足以精确再现场景的视觉效果。Tumblin 和 Rushmeier 正式提出了这个问题，如图 1 所示，并提出了解决这一问题的视觉模型，从此视觉模型就成为了色调映射中的一部分。

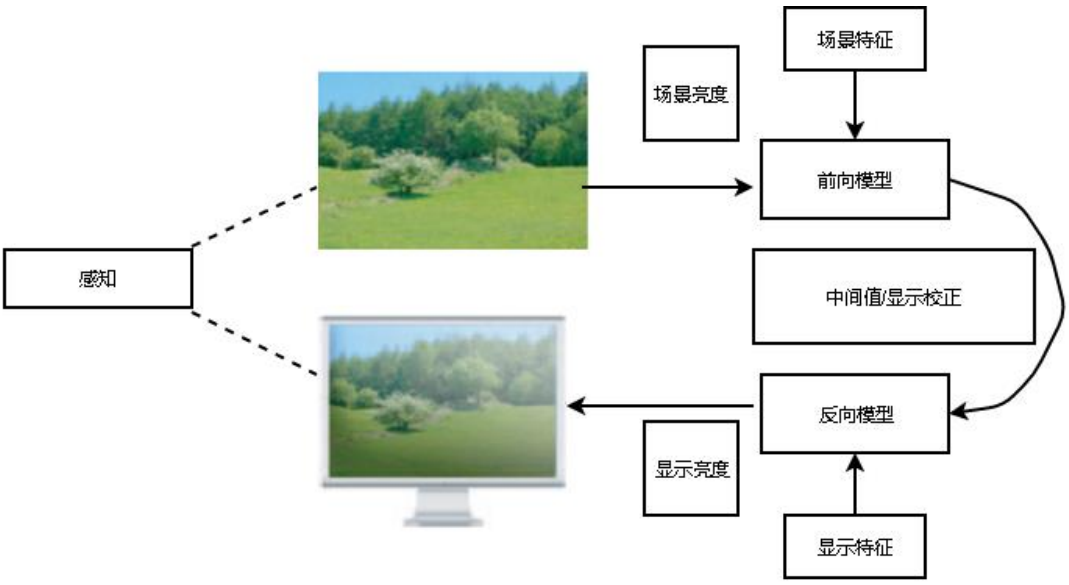

图 1 结合视觉模型的色调映射方案框图

根据我们上一部分介绍的人类视觉感知系统的相关内容，人类对于自然场景的感知是由视网膜上的杆状细胞和锥状细胞捕捉，并进一步通过视觉通路产生视觉信息。这些信号是由多层神经元进行非线性处理，进而形成人对于场景的感知。但是，按照我们在 gamma 校正这一节提到的人眼“视觉亮度非线性”特性，人眼对于亮度的感知与亮度的实际物理辐射亮度之间的关系不是线性映射，这就需要我们在色调映射算法设计的过程中，对这种非线性映射做出相应的处理。

自然场景有着亮度范围非常大的光照条件，从夜晚几乎没有光照、到艳阳天亮度很高的场景。如果在显示器上显示时，只是简单的将真实场景的整个亮度域线性压缩到显示器的范围内，由于“视觉亮度非线性”，图像会在明暗两端损失大量细节，人眼无法感知这一部分的细节变化，这不是场景重现希望得到的效果。人眼是如何解决这个问题的呢？当我们从一个很亮的区域，进入一个很暗的场景时，眼睛所看到的场景会出现一

段时间的模糊，看不清，接着才会变得清晰。这种现象产生的原因是：实际上，虽然人眼能够分辨的亮度域很大，但是这个亮度域中的所有亮度同时出现在一个场景中时，人眼也无法感知其中的全部细节。而人眼的处理方式是：既然无法感知整个亮度域的细节，那么就只关注局部的亮度范围。人眼通过瞳孔的调节，找到场景中亮度的主要分布范围，人眼会对于这个亮度范围内的亮度敏感，能够分辨其中的细节，而对于其他亮度，在人眼的感知中就是过于明亮或者过于阴暗，人眼就无法感知其中的细节。色调映射如前所说，也需要解决这个问题。不同的色调映射算法选择的处理方式有所不同。

色调映射技术可以使用以下的方式做一个统一的定义：

$$f(I): I_i^{w*h*c} \rightarrow D_o^{w*h*c} \tag{1}$$

其中 f 代表的是色调映射算子，I 表示待操作的图像，w，h 代表图像的宽和高，c 代表图像的通道数，通常为 RGB 三通道。同时，因为色调映射算子此时只对亮度进行操作，而不改变颜色，所以上式可以简化为：

$$\mathrm{f(I)} = \begin{cases} L_d = f_L(L_w): I_i^{w*h} \rightarrow [0, 255] \\ \begin{bmatrix} R_d \\ G_d \\ B_d \end{bmatrix} = L_d \left(\frac{1}{L_w} \begin{bmatrix} R_w \\ G_w \\ B_w \end{bmatrix} \right)^s \end{cases} \tag{2}$$

这里$s \in (0,1]$代表饱和度因子，用来降低色调映射后的饱和度，因其在色调映射过程中饱和度通常会增加。操作完成后，通常还会对每个颜色通道进行 Gamma 校正，使它的色度范围固定在[0,255]范围内。

从上面的式子我们可以发现，色调映射算法的关键，就是设计出适合的色调映射因子（Tone Mapping Operator）。色调映射算法的区别，本质上是设计并使用了不同的色调映射因子。

图 2 展示了一个场景的 HDR 图片，该图片在 SDR 显示器上显示时需要进行一些本地处理。当我们对该 HDR 图片进行全局色调映射算法处理后，我们可以得到左侧图片，我们可以看出得到的图片在暗处细节损失严重，很难看清其局部细节；因此我们对 HDR 的图片同时应用全局色调映射算法和局部色调映射算法相结合的方法来渲染正确的图像，也就是图 6.1 中右侧的图片，我们可以看到不论是明亮处的内容、还是暗部细节都可以很好的呈现出来，而单独使用全局映射算法，阴影部分则无法清晰的显示出来。

图 2 HDR 场景示例：左侧使用全局色调映射，右侧使用全局和局部色调映射

尽管市场上已经开始逐渐出现支持 HDR 技术的显示器和电视等，其显示能力已经越来越接近真实世界中的场景，这使得新出现的显示器能够在少量压缩 HDR 的动态范围的情况下显示 HDR 内容。但是，色调映射在 HDR 技术的普及上仍然非常重要，该技术可以很大程度上增强视频内容的效果。

6.1.1 色调映射中的颜色映射

在色调映射过程中，如我们之前所述，在色调映射的过程中，不仅仅图像亮度的范围减小了，图像的颜色范围也减小了。在色调映射的过程中，可以通过对三个通道进行统一的操作，进而将图像的亮度进行适当的映射；也可以分别对单个通道进行不同的操作，进而在对亮度进行映射的同时，也对图像的颜色进行映射。

由于分别对图像的三个通道进行操作，会造成颜色的饱和度降低，色调映射过程中，通常的做法是先对图像的亮度进行映射，然后再通过原图像的信息，获取新的颜色信息(SCHLICK)。也就是说，色调映射过程分为两个步骤，（1）亮度映射，（2）颜色映射。

颜色映射的方式较为统一，通常使用如下式子的方式进行映射：

$$c = \left(\frac{C}{L}\right)^{s} T \tag{3}$$

其中，C 和 c 分别是色调映射前和色调映射后的颜色。L 是 HDR 图像的亮度，T 是色调映射后对应的亮度值。s 是用来调节饱和度的参数，当 s<1 时，颜色的饱和度可以得到调节。

因此，大多数的色调映射算法，都是通过这样的方式进行颜色的映射。色调映射算法的主要内容，也就变为了如何设计更好的亮度映射算法。在本节的后续部分当中，我们提到的色调映射算法，如没有额外叙述，都将表示亮度映射算法。

6.1.2 色调映射算法流水线

不同的色调映射算法有不同的关注点，也有不同的算法设计思路，想要总体地抽象概括全部的色调映射算法的性质与设计思想是十分困难的。但是，随着色调映射算法的研究逐渐成熟与深入，现在的色调映射算法通常都遵循着一种算法设计流水线。现在的大多数算法的不同，也只是体现在流水线中不同步骤的实现方式存在差异，或者额外添加了一些新的限制，但是算法的主要部分，依然遵循着这样一种流水线。在本小节当中，我们就来逐步分析与介绍这种色调映射算法流水线。

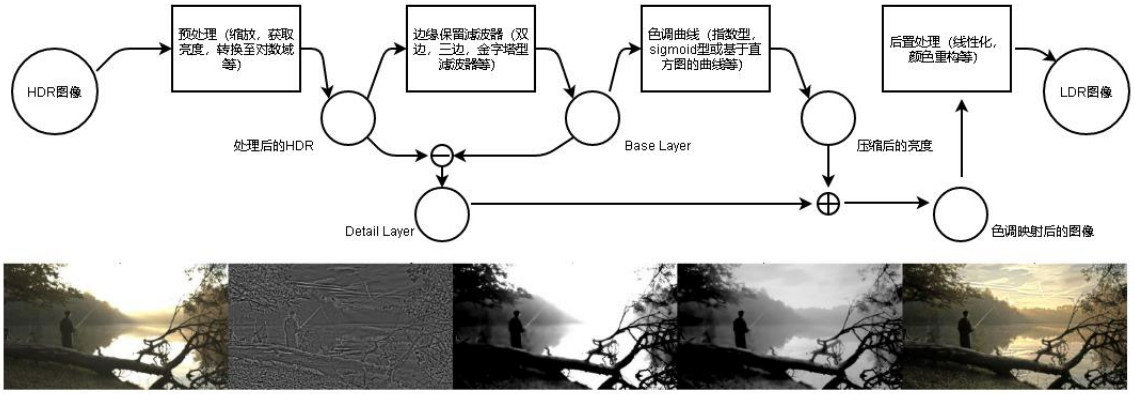

图 3 色调映射算法流程图

这个流水线接收 HDR 图像，输出 LDR 图像。首先，先对输入的 HDR 图像进行预处理，主要包括测算图像的亮度信息，并将亮度信息转换到对数域。接着，处理后的图像通过边缘保留滤波器（例如双边，三边，金字塔型滤波器），提取出图像的亮度信息，也就是图中的 base layer。然后从处理后的图像中减去 base layer，就可以得到 detail layer，也就是图像的边缘信息，或者称之为图像细节。也就是说，第二步操作，将图像的亮度信息与图像细节分开，以供后续步骤进行不同的操作。再之后，第三步，提取出的亮度信息通过不同的 tone curve（色调曲线）进行压缩，或称之为映射，进而映射到 LDR 图像的亮度域之中。最后，将新的，映射后的亮度与之前提取出的 detail layer 叠加，并通过前一节提到的方式，将图像的颜色还原添加，就得到了最终的 LDR 图像。

色调曲线

色调曲线以及局部处理（Local Processing）是色调映射算法设计的核心问题。在上面的色调映射算法流程图中我们提到，提取出的亮度信息，要通过色调曲线，映射到 LDR 图像的亮度域当中。本小节，我们就来看一下什么是色调曲线，以及色调曲线部分有哪些重要的内容。下一小节，则主要介绍局部处理。

在色调映射算法中，最核心的问题就是如何将亮度从 HDR 亮度域映射到 LDR 亮度域。流水线中的其他操作与处理，实际上都是在为这个核心目的服务。在大多数情况下，HDR 到 LDR 的亮度映射，是通过一种连续函数-亮度曲线实现的。

首先，我们公式化地描述色调映射问题。色调映射问题，就是将输入的 HDR 图像 $\mathrm{L} \in R^{+}$映射为 LDR 图像的像素值$\mathrm{l} \in N_d = \{0, 1, 2, \ldots, 255\}$。转换的方式我们记做$\mathrm{M}: \mathrm{L} \rightarrow \mathrm{l}$。L 亮度的计量单位或者存储在图像中的亮度值的计量单位不会影响算法的设计，只要这些亮度与记录的对应场景的亮度之间是线性相关的即可。但是，不同的色调映射算法的设计存在不同，有些算法直接将 L 的亮度值映射为 0-255 的像素值，而有些则将 L 的

亮度值映射为对应的亮度信息 T，然后由显示设备将亮度信息 T 映射为 LDR 图像的像素值。针对这种情况，我们给出色调曲线的定义：我们将色调曲线记做$V: L \to T$，其中$T \in R^+$。也就是说，色调曲线是$R^+ \to R^+$的映射，并且是一段连续的映射，它实现了亮度域的压缩转换。而为了保证算法的完备性，色调映射算法还应当在色调曲线之后，补充亮度信息 T 到 LDR 图像像素值 I 的映射。

色调曲线函数 V 的设计取决于一些自定义的参数$\alpha = \{\alpha_1, \dots, \alpha_{K_\alpha}\}$，例如整体亮度，对比度，饱和度；同时，也取决于一些图像的数据$\phi(L) = \{\phi_1(L_1, \dots, L_N), \dots, \phi_{K_\phi}(L_1, \dots, L_N)\}$。图像数据可以包括图像均值：$\phi = \mu = \frac{1}{N}\sum_s L_s$，图像的最大，最小值以及图像的直方图$\phi_i = p_L(L_i)$。

利用α与ϕ，我们就可以给出空间不变的像素级的色调曲线的定义：$T_s = V(L_s;\ \phi(L);\ \alpha)$。其中 s 表示图像中的像素。

例如，一个简单的曝光补偿可以描述为$T_s = V(L_s;\ \phi(L);\ \alpha) = \alpha L_s$。另一个例子是 Sigmoid 曲线，它可以适应性地调节图像整体亮度，假设输入图像的均值$\phi = \mu$，此时的色调曲线就可以记做：$T_s = V(L_s;\ \phi(L);\ \alpha) = \frac{L_s^\alpha}{L_s^\alpha + \phi^\alpha}$。

上述的两个例子实际上就是简单的色调映射算法，它们的效果虽然并不好，但都实现了亮度域从 HDR 到 LDR 的转换。

局部适应

在上一节中提到，色调曲线可以压缩图像的动态范围，或者说亮度范围，但是，在压缩的过程中，会使得图像整体的对比度出现损失。由于全局对比度在亮度被压缩到较小的亮度范围时不可避免的会出现损失，提升图像的品质就可以从保持图像的局部对比度，或者说细节入手。

我们同样公式化地描述局部对比度。类似于色调曲线中提到的$\phi(L)$，我们使用一系列参数$\xi_s(L)$，来描述局部对比度。这些参数$\xi_s(L)$取决于图像 L，并且它们的取值取决于像素 s 周围的领域Ω_s：$\xi_s(L) = \xi\left(L_{\Omega_s}\right) = \left\{\xi_1\left(L_{\Omega_s}\right), \dots, \xi_{K_\xi}\left(L_{\Omega_s}\right)\right\}$。通过$\xi_s(L)$来获取像素点 s 附近的局部特征，色调映射算法可以在空间上自适应的维持或者增强局部对比度。局部的色调曲线现在可以记做如下形式：$T_s = V(L_s;\ \xi_s(L);\ \phi(L);\ \alpha)$

也就是说，利用$\xi_s(L)$，可以有效地获取图像的局部信息。因此，考虑我们色调映射流水线中的思想，我们可以利用$\xi_s(L)$获取图像的 base_layer，$B_s = \xi_s(L)$，以及 detail_layer，$D_s = L_s - B_s$。利用色调曲线来处理 base_layer，然后将 detail_layer 加回图像中，由此来保持图像的局部对比度：

$$\begin{aligned} T_s &= V(L_s;\ \xi_s(L);\ \phi(L);\ \alpha) \\ &= \tilde{V}(B_s; \phi(L);\ \alpha) + D_s \\ &= \tilde{V}(\ \xi_s(L);\ \phi(L);\ \alpha) + L_s - \xi_s(L) \end{aligned} \quad (4)$$

这里的 V 表示局部色调曲线，而 V 可以被分解为两个部分，一个是用于处理全局图像对比度 B 的全局色调曲线，第二个是细节 D，用于保留细节到最后。

6.2 图像色调映射算法

图像色调映射算法可以分为全局（空间不变）算法和局部（空间变化）算法，他们的建模方式的区别在于部分算法只基于图像全局特性进行调整，另外一部分算法结合图像的全局和局部的视觉特性进行不同的调整。全局色调映射算法对整幅图像的所有像素点采用相同的转换函数，也就是说整幅图像中的所有相同的像素点，在转换后的像素值也是相同的，它们可以是幂函数、对数函数、Sigmoid 或者是基于图像内容的函数。局部色调映射算法在图像中不同的空间区域采用不同的转换函数，这种情况下，映射前颜色相同的区域，在映射后的颜色可能不同，这与其所在位置和周围的像素点值有关。总的来看，全局色调映射算法相对于局部算法来看，有着更快的速度，因局部算法有着更高的计算复杂度。全局算法中，查找表可以更快的处理图片，因此更适合用于照相机中和视频处理。

全局色调映射方法适用于动态范围与显示设备支持的动态范围相近的场景，或者更低。当场景的动态范围远远超过显示设备的显示能力时，全局色调映射方法过度压缩色调范围，导致对比度和可视细节上的无法避免的损失。因此，除了全局色调映射算法外，我们还需要引入局部处理使其具有更好的视觉效果。局部处理算法允许增加局部对比度，从而提高图像某些部分的细节可见性，而全局处理算法将图像的整体动态范围缩放到显示设备支持的动态范围。

6.2.1 全局色调映射

使用全局色调映射算法，就意味着对整幅图像的所有像素点采用一个相同的映射函数来进行处理。这样做的好处是，对所有的像素点进行相同的操作可以保留整幅图片的全局对比度。映射函数有时可能会对输入的图像先做一次处理来计算得到一些重要的全局信息，然后使用计算得到的全局信息来进行色调映射。色调映射算法中常用的全局信息包括最大亮度、最小亮度、对数平均值和算数平均值等。为了增强算法的鲁棒性并减少结果中的异常值，这些全局信息一般使用百分数计算，尤其是最大亮度和最小亮度信息。在时域上应用全局映射算法也是十分直观的，在多数情况下我们需要暂时过滤计算完成的图像数据，因为这可以避免因为序列中的结构不连续导致的闪烁问题。全局映射算法的主要缺陷在于因为操作过程中使用的是图像的全局信息，因此无法获得图像的全局对比度以及原始 HDR 图像中良好的细节。

全局色调映射方法的设计通常遵循两种不同的思路：第一种是从传统的数字图像处理的方法出发，通过计算得到图像的某些全局信息，来对像素点进行操作；第二种是从人类视觉感知系统的某些特性出发，通过人类视觉感知系统的某些结论，来设计色调映射算法。基于数字图像处理方法是最先应用到色调映射算法当中的，而随着人类对视觉的研究越发深入，基于人类视觉感知系统的色调映射方法才逐渐进入人们的视野。通常情况下，基于人类视觉感知系统的色调映射算法得到的结果要更加优秀。

本节中，我们将首先按照介绍两种基于数字图像处理的算法：简单映射算法以及直方图校准算法。接着，我们将介绍两种主要的基于人类视觉感知系统的算法：视觉适应模型算法以及时变视觉适应算法。这些方法各有优劣，并且已经很少在实践中使用了，但是这些方法的设计思路以及应用到的原理，仍然能够带给我们一些启发，因此本节介绍的这些算法中，我们不用过于关注具体实现以及数学推导，我们需要关注的重点是这些算法的设计思路与想法。

1. 简单映射算法

我们介绍的第一种基于数字图像处理的色调映射算法称为简单映射算法。全局色调映射技术是将 HDR 图像的每一个像素点通过同一个映射因子映射到 LDR 图像的范围内。因此，本质上来说，色调映射就是两个图像空间的映射，而最简单也是最直观的映射方式就是使用一些基本初等函数进行映射。基于基本初等函数的色调映射算法就被称为简单映射算法。

简单映射算法中使用的基本函数通常为线性函数、对数函数和指数函数。尽管这些基本函数可以简单而快速的实现整个映射过程，但他们却无法将图像的动态范围准确的压缩。线性曝光是显示 HDR 图像的一种非常直接的方法，初始图像被乘以一个因子 e，就像数码相机处理曝光度的方法一样：

$$L_d(x) = eL_w(x) \tag{5}$$

使用者可以根据关注的信息来选择 e 的取值，当$e = \frac{1}{L_{w,max}}$时，上式被称为标准化形式，此时得到的结果是图像会变得很暗。当 e 的取值能使图像中有着最多的曝光良好的像素点时，上式可以被称为自动曝光形式。但是，一个简单的线性变换无法很好的完成动态范围的压缩任务，因为线性变换只能良好的显示其中一小部分的信息。

对数映射则是利用对数函数对 HDR 图像进行映射操作，对数映射将会以 HDR 图像中的最大值为标准，将整幅图像非线性地映射到[0,1]区间内，其映射函数可以写为：

$$L_d(x) = \frac{\lg(1+qL_w(x))}{\lg(1+kL_{w,max})} \tag{6}$$

其中$q, k \in [1, +\infty)$是由使用者定义的参数，这两个参数可以决定映射算法的具体表现。

指数映射是利用指数函数对 HDR 图像进行映射操作，指数映射将图像各像素点的值通过由指数函数构成的函数映射到[0,1]区间内，具体的映射函数定义如下：

$$L_d(x) = 1 - e^{\left(-\frac{qL_w(x)}{kL_{w,H}}\right)} \tag{7}$$

同样，q，k 也是使用者需要调节的参数。使用上述几种方法得到的效果如图 4 所示。

图 4 简单映射方法的结果图示例：(a)标准化形式；(b)自动曝光形式；(c)对数映射，其中 q=0.01, k=1；(d)指数映射，其中 q=0.1, k=1。

对数映射和指数映射都可以在处理出于中段动态范围的内容时，获得很好的效果。但是，这两种方法在计算整幅 HDR 图片时却效果欠佳，这将导致图像过于明亮或过于

阴暗，全局对比度也会有失真并且图像细节不够自然。

2. 直方图校准

最后一种基于数字图像处理的算法是直方图校准算法，在这种算法中，也加入了一些人类视觉感知系统的理论应用。Larson 将传统的直方图均衡技术进行了修改和调整，使它能够用于色调映射技术上，Larson 同时也在其映射算法中模拟了人类视觉系统的一些特性。首先，这种方法要计算输入图像的灰度直方图 I，并在对数域内使用二进制数 n_{bin}表示。Larson 通过实验证明，最多需要 100 个二进制数就足够准确地表示结果，此时，累计直方图 P 表示如下：

$$\mathrm{P(x)} = \sum_{i=1}^{x} \frac{I(i)}{T}, T = \sum_{i=1}^{n_{bin}} I(i) \tag{8}$$

其中 x 是二进制数，这里需要注意的是累计直方图是一个积分形式，而灰度直方图是它在适宜尺度下的导数：

$$\frac{\partial P(x)}{\partial x} = \frac{I(x)}{T\Delta x}, \Delta x = \frac{\lceil \log(L_{w,max}/L_{w,min}) \rceil}{n_{bin}} \tag{9}$$

随后，灰度直方图需要均衡化，传统的均衡化对比度方法如下：

$$\log\big(L_d(x)\big) = \log\big(L_{d,min}\big) + P\big(logL_w(x)\big)\log(L_{d,max}/L_{d,min}) \tag{10}$$

这种操作因为只利用了很少的点进行区域内的色度域压缩并用最中间的点进行拓展，所以会引起图像中一大片区域的对比度失真，因此他们使用了一种直接的方法，可以用下式表示：

$$\frac{\partial L_d}{\partial L_w} \leq \frac{L_d}{L_w}$$

综合上面的三个式子，我们可以得到：

$$e^{\log(L_d)} \frac{f(\log(L_w))}{T\Delta x} \frac{\log(L_{d,max}/L_{d,min})}{L_w} \leq \frac{L_d}{L_w} \tag{11}$$

上式可以简化为：

$$\mathrm{f(x)} \leq \mathrm{c}, \mathrm{c} = \frac{T\Delta x}{\log(L_{d,max}/L_{d,min})} \tag{12}$$

因此，当上式的条件无法满足时，对比度的失真就产生了，解决方法就是截断 ***f(x)***，不过这种操作需要迭代的进行，以避免改变 T 和 c。这种算法引入了一些模拟人类视觉系统的机制，例如对比度、锐度和颜色敏感度等，这些都是受启发于 Ferwerda 之前的工作。

总的来说，这种色调映射算法提供了一种改进 HDR 图像的直方图均衡方法，可以提供效果更好的动态范围压缩以及良好的整体对比度。

3. 视觉适应模型

Ferwerda 提出了一种视觉适应模型色调映射算法，这种算法根据人类视觉系统的适应机制，对人类视觉系统的部分方面进行建模，如阈值可见性、颜色感知、视觉灵敏度和时间灵敏度等，通过分段函数来实现对于视杆细胞（T_p）和视锥细胞（T_s）的建模，模型函数可以定义为：

$$\log_{10} T_p(x) = \begin{cases} -0.72 & \log_{10} x \le -2.6 \\ \log_{10} x - 1.255 & \log_{10} x \ge 1.9 \\ (0.249 \log_{10} x + 0.65)^2 - 0.72 & otherwise \end{cases} \tag{13}$$

$$\log_{10} T_s(x) = \begin{cases} -2.86 & \log_{10} x \le -3.94 \\ \log_{10} x - 0.395 & \log_{10} x \ge -1.44 \\ (0.405 \log_{10} x + 1.6)^{2.18} - 2.86 & otherwise \end{cases} \tag{14}$$

映射过程是对每个颜色通道使用一个简单的线性方程来重建光照效果，同时加入了一个用于适应暗条件的消色差项，因此方程可以表示为：

$$\begin{bmatrix} R_d(x) \\ G_d(x) \\ B_d(x) \end{bmatrix} = m_c(L_{da}, L_{wa}) \begin{bmatrix} R_w(x) \\ G_w(x) \\ B_w(x) \end{bmatrix} + m_r(L_{da}, L_{wa}) \begin{bmatrix} L_w(x) \\ L_w(x) \\ L_w(x) \end{bmatrix} \tag{15}$$

其中L_{da}表示显示屏幕的亮度适应范围，L_{wa}表示图像的亮度适应范围，m_r, m_c是两个取决于分段函数的权重因子，通过以下等式定义：

$$m_r(L_{da}, L_{wa}) = \frac{T_p(L_{da})}{T_p(L_{wa})}, m_c(L_{da}, L_{wa}) = \frac{T_s(L_{da})}{T_s(L_{wa})} \tag{16}$$

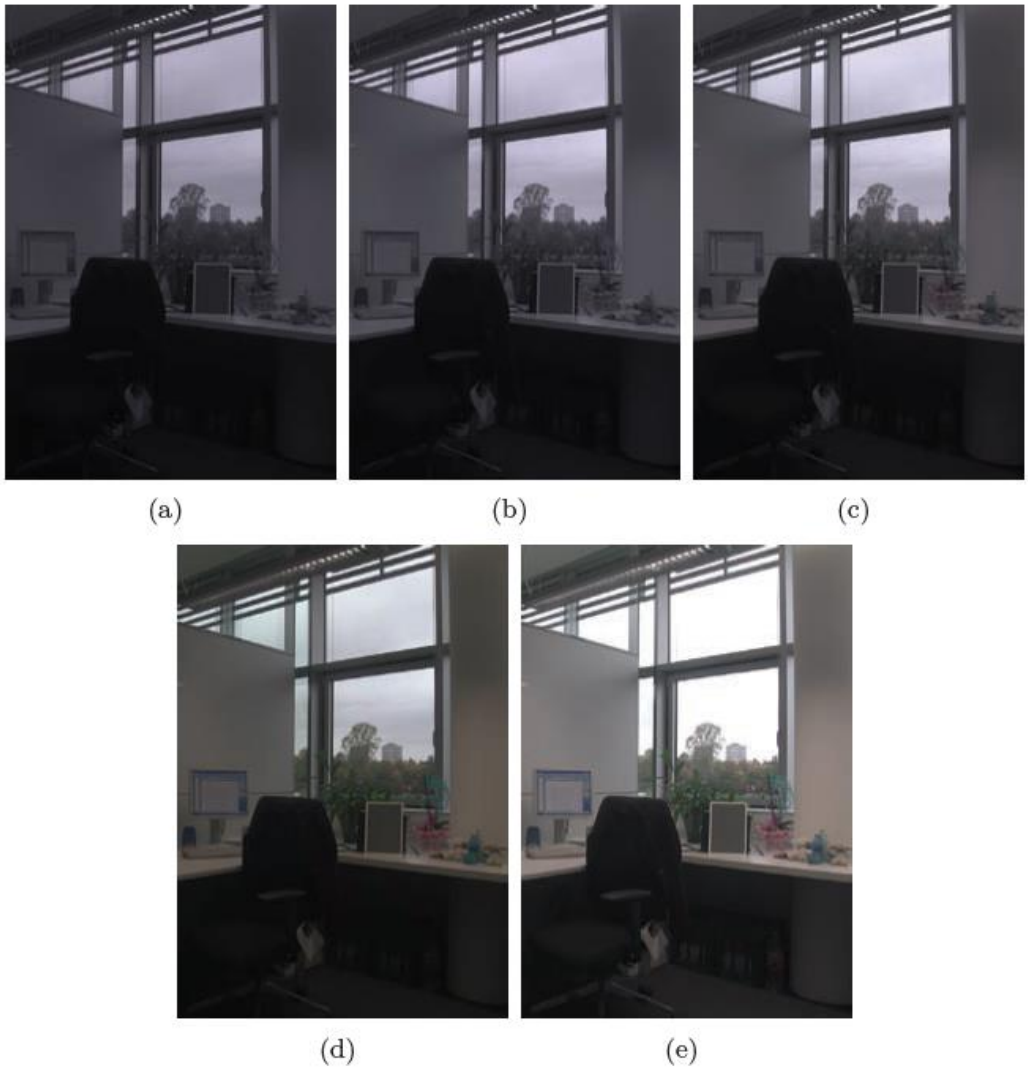

图 5 Ferwerda 的方法下不同平均亮度设定时的结果。(a)0.01 cd/m2；(b) 0.1

cd/m2；(c) 1 cd/m2；(d) 10 cd/m2；(e) 100 cd/m2。

当然，本色调映射方法提供了多种引入人类视觉系统仿真的想法，在缩放动态范围的同时，通过对人类视觉系统部分特性的建模，实现了暗环境细节在 LDR 显示器上显示时的适应能力，但是其动态范围的减少是通过一个简单的线性缩放，因此无法达到更好的动态范围压缩效果。

6.2.2 局部色调映射

相比于全局色调映射因子，局部色调映射因子更能提高色调映射图片的质量，因为局部色调映射因子不仅着眼于全局的对比度构建，也着眼于局部的对比度构建。映射因子 f 通过将正在映射的像素点，以及这个像素点周围的像素点的强度值同时纳入计算，进而得到映射后的图像的像素值。尽管局部色调映射方法在理论上产生的图像效果要好于全局色调映射因子，但是，在局部色调映射算法的实际应用中通常会出现一个问题，进而影响图像的质量。这个问题就是光晕现象。

在局部色调映射算法的设计过程中，如果周围的像素点的选择不够好，进行的映射方法的设计不够好，就有可能在图像某些区域的边缘出现明显的光晕。尽管在一些时候使用者希望认为产生光晕，因为这可以提醒人们去注意某一特定区域，但是，由于光晕的产生无法被控制，并且通常情况下会给图像效果带来不良影响，因此，在实际算法设计过程中，通常都需要考虑如何避免光晕的产生。

局部色调映射算法的理论基础较为复杂。大多数局部色调映射算法都是综合了很多人类视觉感知系统的结论或是基于相关研究而得出的。在本节我们将给出四种经典的局部色调映射算法，我们在了解算法的同时，也应当注意在这些算法中是如何消除光晕的。

1. 空间不均匀缩放

Chiu 最先提出了一种保持局部对比度的方法。这种色调映射因子通过一个像素点周围的其他像素点的均值来衡量这个像素点的亮度。定义如下：

$$L_d(x) = L_w(x)s(x) \tag{17}$$

这里的 s(x)是用来衡量周围像素点局部平均值的测量函数。定义如下：

$$\mathrm{s(x)} = (k(L_w \otimes G_\sigma)(x))^{-1} \tag{18}$$

其中G_σ是一个高斯滤波器，k 是用来衡量最终结果的常数。这种色调映射算法存在的一个问题是如果σ过小，那么产生的图像的对比度就会很低，效果不好；如果σ过大，那么产生的图像中会出现光晕。光晕通常会出现在明亮区域与阴暗区域的交界处，这意味着$\mathrm{s(x)} > L_w(x)^{-1}$。为了减轻这种情况带来的影响，我们将在$\mathrm{s(x)} > L_w(x)^{-1}$时，把 s(x)的值固定为$L_w(x)^{-1}$。在$\mathrm{s(x)} > L_w(x)^{-1}$的点上，s 仍然会有些人为操作的痕迹在里面，主要的表现形式是会有陡坡的出现。一个解决的方法是使用一个 3×3 的高斯滤波器来迭代地平滑 s(x)。最后，该算法再使用一个低通滤波器来掩盖那些引人注意的人为产生的光晕。

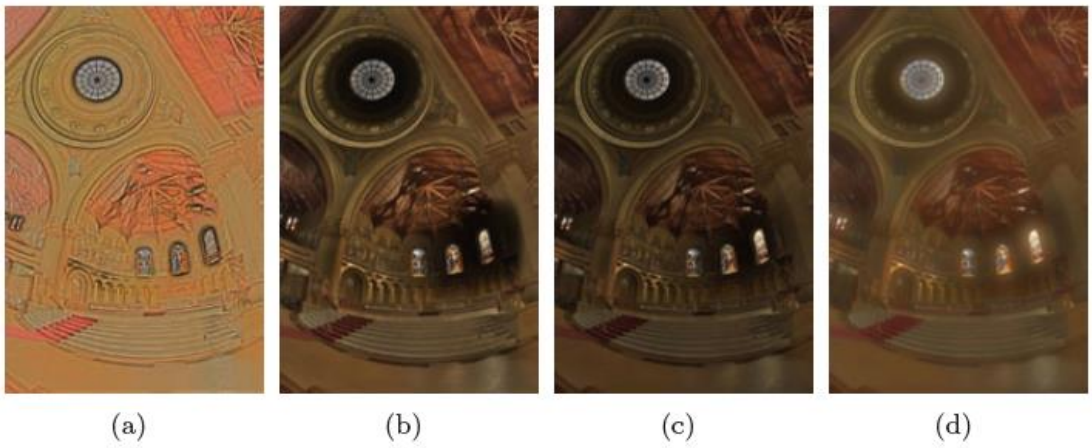

图 6 Chiu 局部色调映射算法的示例。

如图 6 所示，是本方法的结果图示例，其中(a)是$\sigma = 3$的简单算法，可以看出局部信息保留着，但全局信息丢失严重；(b)代表$\sigma = 27$的简单算法，这种情况下局部信息和全局信息都被保留了，但图中的光晕非常明显；(c)是$\sigma = 27$并固定 s 值，光晕有所消减；(d) $\sigma = 27$并有眩光模拟，消除了大量光晕。

这是人们提出的第一个局部色调映射操作因子，但是在减轻光晕带来的影响时，这种操作因子产生的计算代价过于高昂（在平滑阶段大约需要进行 1000 次迭代）。有太多的参数需要去被调整，而且在一系列的操作之后，光晕仍然只能被减轻，而不能被完全消除。

2. 摄影学色调重现

Reinhard 提出了一种基于摄影原理的局部色调映射因子，这种方法模拟了摄影技术中使用了超过一个世纪的 burning&dodge 效应，这种算法灵感来源于 Adams 提出的 Zonal 系统。这种算法的全局分量主要对高亮度的部分进行压缩：

$$L_d(x) = \frac{L_m(x)}{1+L_m(x)} \tag{19}$$

其中L_m是对$aL_{w,h}^{-1}$进行缩放的原始亮度，a 是选定的曝光度，$L_{w,h}$是场景键值的对数平均数估计值。键值主观地认定该场景是明亮的、正常的还是暗的，并且会被用在区域系统中来预测一个场景亮度是如何映射到打印区域中的。注意到在上式中，高亮度被压缩，而其他亮度则是被线性缩放的。但是，上式并不会出现明亮的区域变得更亮，但在摄影中，摄影人很可能会为了强调某些场景而加大曝光，加大对比度。因此，上式可以被修改成下面这种形式：

$$L_d(x) = \frac{L_m(x)\left(1+L_{white}^{-2}L_m(x)\right)}{1+L_m(x)} \tag{20}$$

这里的L_{white}表示会被映射到白色的最小亮度值，默认情况下它与$L_{m,max}$相等，L_{white}作为截断值，会将超过该值的像素点值截断为L_{white}。

(a) (b)

图 7 Reinhard 摄影学色调重现的示例。(a)$\Phi = 4, \alpha = 0.05, L_{white} = 10^6 cd/m^2$；(b) $\Phi = 4, \alpha = 0.05, L_{white}$与窗口亮度接近。

通过之前的叙述可以定义一个局部色调映射算法，具体的操作方式是找到一个最大的没有明显边界的局部区域，这样可以避免产生光晕效应。这样的区域可以通过比较不同尺寸的高斯滤波之后的图片L_m，如果其差别很小甚至趋近于零，那就说明没有明显边界，否则有明显边界，判别方程如下：

$$\left|\frac{L_\sigma(x)-L_{\sigma+1}(x)}{2^\Phi a\sigma^{-2}+L_\sigma(x)}\right| \leq \varepsilon \tag{21}$$

当所有像素点的最大σ都满足上式，也就是都没有明显边界时，全局操作因子就会被更改为局部操作因子，具体如下：

$$L_d(x) = \frac{L_m(x)}{1 + L_{\sigma_{min}}(x)}$$

$$L_d(x) = \frac{L_m(x)\left(L_{white}^{-2} L_m(x)\right)}{1+L_{\sigma_{min}}(x)} \tag{22}$$

式中的$L_{\sigma_{min}}(x)$是图像像素周围最大区域σ_{max}的平均亮度值。摄影学色调重现是一种保留了边界效应，并且避免了光晕效应的局部色调映射算法，除此之外它还有个优势，不需要输入校正后的图片。

6.2.3 频域算法

前面两节介绍的全局与局部色调映射算法可以统一地看做在空间域上，对图像进行色调映射的操作，本节，我们将介绍两种在频域下对图像进行色调映射操作的方法：快速双边滤波算法以及梯度域压缩算法，以供读者参考。

基于频域的算法具有与局部映射算法相同的目标，也就是保持边缘和局部对比度的目标，在频率运算符的情况下，通过在频域下而不是空间域下的计算来实现。这种方法

当且仅当大型特征和细节直接完全分离时，边缘信息和局部对比度信息才能得以保留。因此，频域算法的主要任务，实际上是将图像滤波出高频与低频部分，然后分别进行处理。

1. 快速双边滤波算法

双边滤波（Bilateral Filter）是一种非线性的滤波，可以将图像分离为高频图像，被称为细节层，并将保留边缘的低频图像成为基本层。Durand 和 Dorsey 利用这个特性设计了一个通用且高效的色调映射框架，保留了局部对比度特性。

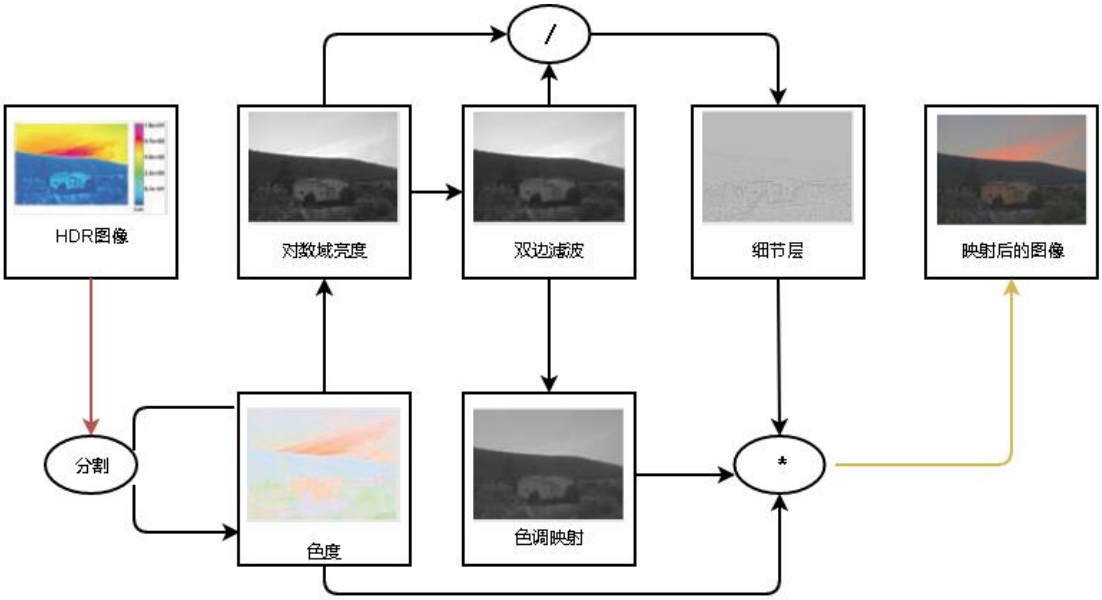

图 8 快速双边滤波算法的流程框图

方法的流程框图如图 8 所示，框架的第一步是将 HDR 图像分解为亮度分量和色度分量，此时用双边滤波器对对数域内的亮度进行滤波，并通过亮度与滤波后的商来计算细节层，滤波后的亮度随后使用一个全局算法进行色调映射。最后，将色调映射的基本层、细节层和色度分量重组，形成色调映射的最终结果。

图 9 使用双边滤波结构与否的效果对比。(a)亮度再现方法的结果；(b)亮度再现方法与双边滤波结构结合方法的效果。

Durand 和 Dorsey 提出了一个使用近似双边滤波和降采样的加速滤波器，然而这种技术已经被后来引入的加速双边滤波器淘汰了，这种加速方法可以应用于任何一种全局色调映射算法。图 9 显示了一种例子，与这种方法相关的一个问题是，光晕并没有被完全消除，因此对该种方法的改进，有使用三边滤波器等方法出现。

6.2.4 近期一些新算法

近年来，随着人们对于图像以及视觉的研究不断深入，色调映射算法的设计也逐渐变得更加复杂，适应性与通用性也逐渐加强。本节，我们将介绍两种近年来表现效果良好的三种色调映射算法：分割色调映射算法，Reinhard 色调映射算法以及自动淡化与加深算法。

1. 分割色调映射算法

最近，一种新的色调映射算法将分割算子的思想纳入其中。强边缘和大多数局部对比知觉位于大均匀区域的边界上。分割算子将图像分割成均匀的段，在每个段上应用一个全局算子，最后合并它们。这种方法的另一个优点是，最小化色域修改，因为在每个情况下，每个片段的线性算子有时都是足够的。

色调映射中的明度感知

Krawezyk 提出了一种基于明度感知锚定理论的色调映射算法，这种理论指出人类视觉系统可以将视野中的最高亮度值也叫锚点，视为白色，这种感知方法受到相对区域的影响。当最高亮度覆盖一小个区域时，它看起来是自发光的。为了将光度理论应用于复杂的图像，Gilchrist 等人提出在可以应用锚定理论的区域中分解图像，被称为框架。

算法的第一步是确定框架，然后在 log10 的对数域下计算图像直方图，使用 k-means 聚类算法确定直方图中的质心G_i，通过基于像素计数的加权平均合并接近质心。为了避免接缝或不连续性，使用软分割生成框架。每个框架定义一个概率函数，确定这个像素是否属于基于质心的框架，如下所示：

$$P_i(x) = \exp\left(-\frac{(C_i - \log_{10}(L_w(x)))^2}{2\sigma^2}\right) \tag{23}$$

其中σ等于两个框架之间的最大距离，$P_i(x)$使用双边滤波平滑处理，以删除局部的小变化，如图 10 所示。一个框架的局部锚点ω_i通过计算框架中亮度第 95%的点来确定。最终色调映射的图片可以被计算为：

$$L_d(x) = \log_{10}(L_w(x)) - \sum_{i=1}^{n} \omega_i P_i(x) \tag{24}$$

色调映射的最终图片示例可以从图 10 看到，算法通过将其与摄影色调再现算法和快速双边滤波算法的结合来比较验证 Gelb 效应，这是一种与明度不愿相关的错觉。试验结果表明，基于亮度的算法可以重现这种效果，算法可以快速直观地进行实施，但在将其应用于动态场景时需要特别小心，以避免出现重影伪像等情况。

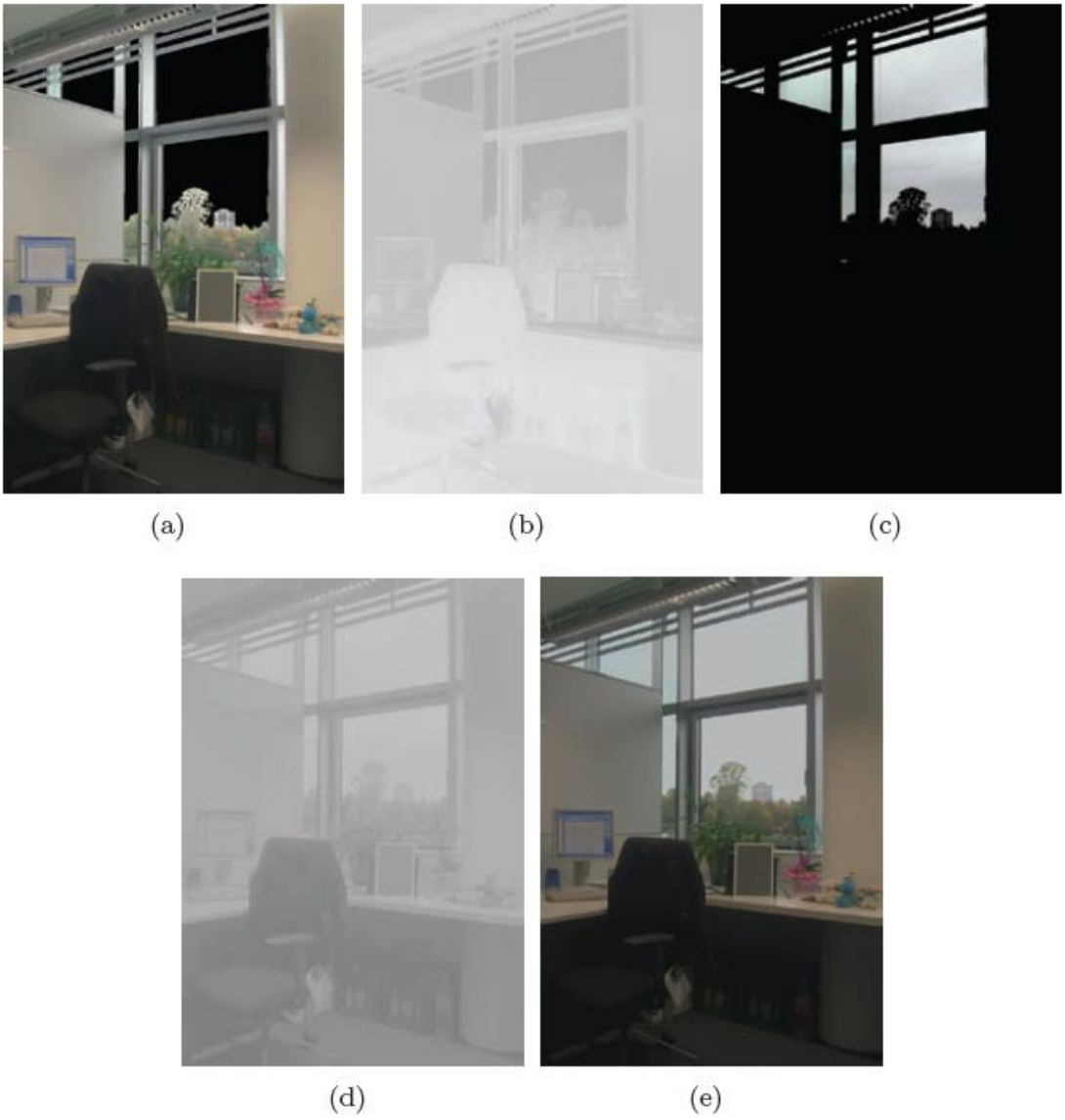

图 10 Krawezyk 色调映射算法结果图示例。其中(a)、(c)框架中使用锚点，(b)、(d)将(a)、(c)的概率分布平滑处理，(e)是通过合并框架得到的最终色调映射后的图像。

曝光融合

在色调映射之前，HDR 图像通常需要由一系列 SDR 图像组合而成，Mertens 提出来一种可以避免色调映射步骤的新方法，该算法的核心概念是合并每次曝光中曝光良好的像素点。

算法第一步是分析每个 SDR 图像以确定在合并时需要使用哪些像素，这是通过计算每个像素的三个度量来实现的：对比度、饱和度和良好曝光度。对比度 C 是指图像中渐变的绝对值，饱和度 S 定义为红色、绿色、蓝色三通道的标准偏差，亮度 L 的良好曝光度 E 确定像素是否以模糊方式良好曝光。

$$\mathrm{E(L)} = \exp(-0.5(L-0.5)^2\sigma^{-2}) \tag{25}$$

这三种度量结合，得到了一个权重和$W_i(x)$，它决定这个像素点在当前曝光度下的重要性：

$$W_i(x) = C_i(X)^{\omega_c} \times S_i(x)^{\omega_s} \times E_i(x)^{\omega_e} \tag{26}$$

其中 i 指第 i 张图像，$\omega_c, \omega_s, \omega_e$是增加度量标准对其他标准影响的指数，N 个权重

图都被归一化处理过，使得它们的总和在每个像素位置处等于 1，以便获得一致的结果。

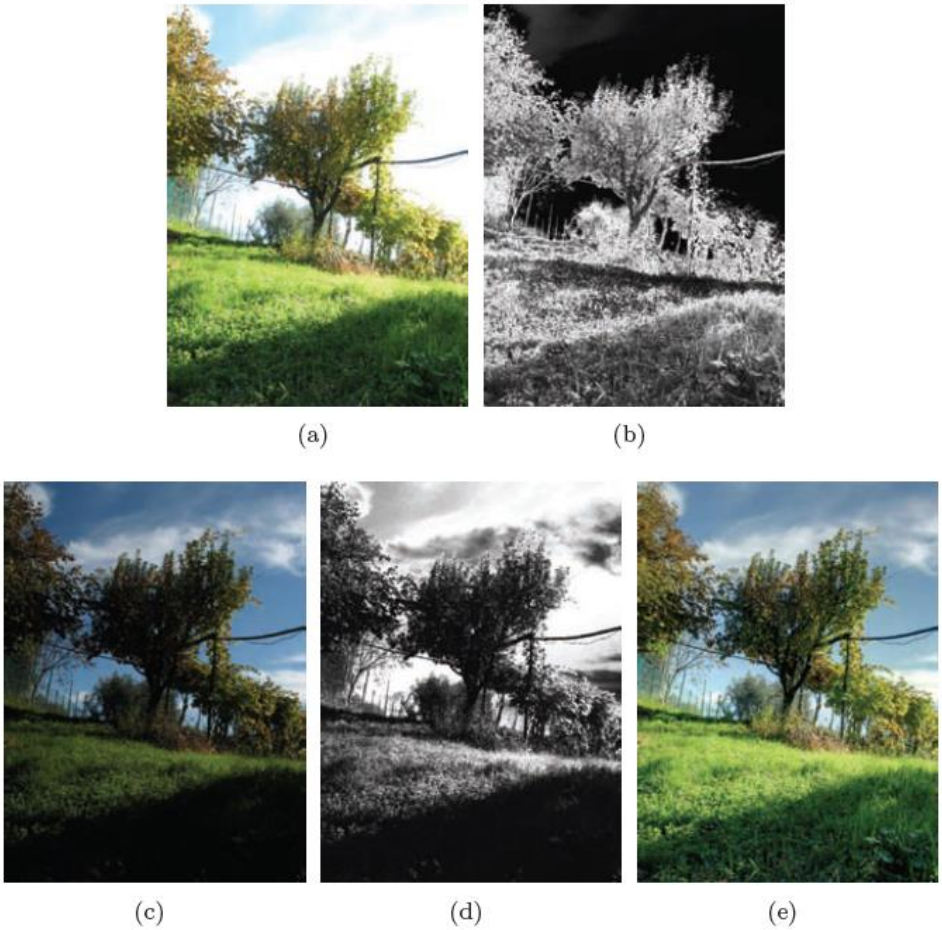

图 11 Mertens 的曝光融合算法示例。(a)是 HDR 图像的第一个曝光度；(b)是(a)的权重图；(c)是 HDR 图像的第二个曝光度；(d)是(c)的权重图；(e)是使用拉普拉斯金字塔融合后的色调映射图像。

分析之后，曝光结合在最终图像中，为了避免接缝和不连续性，图像混合使用拉普拉斯金字塔。虽然如用运算符 G 表示的那样在高斯金字塔中分解权重，但是如运算符 L 所示，曝光图像I_k被分解为拉普拉斯金字塔，因此混合计算可以定义为：

$$L^l(I_d)(x) = \sum_{l=1}^{n} L^l(I_l)(x)G^l(W_l)(x) \quad (27)$$

其中 l 表示拉普拉斯或高斯金字塔的第 l 层。最终$\mathrm{L}(I_d)$消掉了，得到了色调映射后的图片I_d，如图 11 所示。

该算法的主要优点是使用者不需要生成 HDR 图像，此外，它还可以最大限度地减少传统色调映射算法中可能出现先的颜色偏差，这是因为曝光良好的像素是在没有应用实际压缩功能的情况下拍摄的，只是线性关系。

2. Reinhard 色调映射算法

Reinhard 算法是一种基于区域系统的算法。在具体的操作过程中，算法会利用到一些摄影领域的技术，主要的应用就是利用区域系统的基本概念模型来对色调重构进行选择。操作过程中，我们首先会对图像进行缩放处理，就像照相机的曝光那样。然后，如果需要的话，我们会对图像进行自动淡化或加深处理，这样可以实现动态范围的压缩。

首先要做的是如何基于场景的关键值来设定输出图像的色调范围。与很多其他的色调映射方法类似，我们也认为对数平均亮度是场景关键值的有效近似。这个值$\bar{L}_w$可由下式得到：

$$\overline{L_w} = \frac{1}{N}\exp\left(\sum_{x,y}\log(\delta + L_w(x,y))\right) \tag{28}$$

$L_w(x,y)$是真实世界中（x，y）这个像素点的亮度值。N 是图像中的像素总数。δ是用来避免奇异化的一个很小的量。如果场景的关键值是正常的，那么我们会把亮度映射到显示图形的中间灰度去，或者 0.18.从 0 到 1 进行缩放。如下式：$\mathrm{L(x,y)} = \frac{a}{\overline{L_w}}L_w(x,y)$

$\mathrm{L(x,y)}$是缩放后的亮度，*a=0.18*。对于低关键值和高关键值的图像我们允许使用者将对数平均值映射到 a 的不同取值上。一般地，*a* 向上会取 0.36，0.72；向下会取 0.09，0.045。

上面的式子存在的一个问题是，虽然大多数场景的主要部分都属于正常的亮度范围内，但是仍然有少数高亮的部分以及天空部分属于高亮度范围。在传统摄影技术中这种问题是通过同时压缩高低亮度部分来完成的。

但是，在现在的图像技术中，我们不再继续使用这种同时压缩高低亮度的“s”型的映射曲线，而是主要对高亮度部分进行压缩。一个符合这种要求的简单操作因子如下：

$$L_d(x,y) = \frac{L(x,y)}{1 + L(x,y)}$$

高亮度部分的缩放比约为$1/L$，而低亮度部分则为 1。这种方法在两种缩放方式之间达成了很好的协调。而且这个式子可以保证将所有的亮度都映射到可以显示的亮度范围内。但是，并不是所有时候都需要这样的效果，上式可以经过适当的修改，然后可以让高亮度部分在可控范围内得到加强：

$$L_d(x,y) = \frac{L(x,y)\left(1+\frac{L(x,y)}{L_{whtie}^2}\right)}{1+L(x,y)} \tag{29}$$

L_{white}是会被映射为白色的最小的亮度值。这个式子可以看做前面的操作因子与线性映射的混合。

对于大多数 HDR 图像来说，上面的方法提供的动态范围压缩方式已经足够保存低对比度区域的细节了。但是对于那些有着非常高的动态范围的图像来说，这种方法不足以保存它们存在于高动态范围的细节，因此对这些图像来说，淡化与加深算法是需要的。

6.3 视频色调映射算法

视频色调映射算法与图像色调映射算法存在很大的不同。视频色调映射算法实际上是对一段连续的帧进行色调映射，与图像色调映射算法相比，两者之间最大的不同是视频色调映射算法需要考虑到，并且准确地处理相邻帧之间的相关性，即考虑时域特性。如果这种帧间的相关性，或者换言之，这种时域的联系没有被考虑到，或者设计的算法不能准确处理这种时域联系，那么，在经过色调映射之后，视频就会出现一系列的问题，例如帧间闪烁现象。这种情况下，视频的人工痕迹就会非常明显，视频的质量与观看体验就会大幅下降。而这种时域联系并非是视频色调映射算法面临的唯一问题。由于视频处理要处理大量的帧，因此计算的效率也是十分重要的一个方面。另外，由于不同的 HDR 获取设备使用不同的方式获取 HDR 视频，因此这些视频本身也存在差异。所以，想要设计出适应多种情况的高质量的视频色调映射算法，需要综合考虑多方面的情况。

6.3.1 背景

尽管 HDR 视频的获取技术在 2010 年左右才相对成熟，视频色调映射算法早在这之前 15 年，就已经开始出现了。早期的视频色调映射算法都会涉及出固定的时域处理算法或者模型来处理视频的时域特征，但是这是不全面的。视频色调映射算法的最终目的，是建立一套自适应的模型，或者算法，来解决所有视频的时域问题。但是，由于早期缺乏高质量的 HDR 视频源，因此在当时无法精确地判断一个视频色调映射算法的性能。

随着 HDR 视频获取技术的发展以及新的 HDR 视频采集设备的使用，视频色调映射算法的性能可以得以检验。经过相关的调研 (EILERTSEN G.)，我们发现，大多数的视频色调映射算法都存在相同的几类问题：人工痕迹过重，对比度缺失以及细节缺失。因此，在 2010 年之后，视频色调映射算法的设计已经逐渐转变为针对具体存在的问题，做出相应的调整与完善。而视频色调映射算法的设计目标，也被明确为：设计出与视频内容无关的高质量色调映射算法。

时域特性

Boitard 曾经提出过视频色调映射过程中可能产生的时域问题的类型，其中主要包括：帧间闪烁，时域对比度缺失以及物体不连贯 (BOITARD R. C. R., 2014)。这些问题当中，我们主要关注全局与局部的闪烁现象以及鬼影现象，因为这三个问题，是视频色调映射过程中最常出现的时域问题。而其他的问题，例如时域对比度缺失以及物体不连贯，相较之下就容易处理得多，因为人们对于这些现象相对比较习惯。只要物体与对比度的变化是平滑的，那么，这些变化在人们看起来就是自然的，可以接受的，也就是说，这些问题并不会影响到人们的观看体验。而前面提到的三个主要的问题，全局与局部的闪烁现象以及鬼影现象，就会给人们带来很差的观看体验。因此，视频色调映射算法需要注重处理全局与局部的闪烁现象以及鬼影现象。

假设 V：L->T 代表一个由 HDR 视频到 LDR 视频的映射，如果这个映射$V = V(L_s; \phi(V))$，其中L_s表示映射的输入内容是 HDR 视频，$\phi(V)$是通过某些方式得到的图像与视频信息。在这种情况下，这个映射 V 就是一个与图像内容相关的映射。由于$\phi(V)$在部分帧与帧之间会剧烈的改变，这样就会影响到色调映射算法的效果。体现在输出的视频上，就是会出现帧间闪烁的情况。

对于视频时域特性的处理，第一种方式是进行像素级的改变，也就是说对于图像的操作实直接对像素值进行改变。如我们在图像色调映射算法中所说的那样，全局色调映射算法会对所有的像素点使用同样的色调曲线进行映射。因此，想要在全局处理中保持帧间变化的平滑性，只需要将连续的帧的信息通过一个时域低通滤波器，其中使用的核记做G_σ。

$$\tilde{\phi}(L_t) = \sum_{r=t-d}^{t+d} G_\sigma(\phi(L_r)) \tag{30}$$

从上面的处理中我们能够发现，这种时域低通滤波，是一种非因果的操作。如果希望保证算法的因果性，我们可以只对时间 t 之前的帧，进行低通滤波。

而对于局部色调映射算法来说，情况就更加复杂。如果局部色调映射算法使用了一些局部的图像信息$\xi_s(L)$，并且没有适当的处理这些信息，那么，就会造成图像在某些区域剧烈地改变，这就会造成局部视频闪烁的发生。而如果向上面的式子所述的那样，直接对每个像素进行低通滤波，去除全局闪烁，生成的视频往往会产生鬼影现象。一种可以选择的改变方式是不直接对图像信息进行滤波，而是对色调映射后的亮度信息进行低

通滤波：$\widetilde{T_s} = (G_\sigma * l)_s$。这种操作既可以保证时域相关性，又可以尽量避免噪声干扰，但是，这却会产生动作的模糊。

为了使像素级的时域滤波避免动作模糊的影响，可以对滤波器的核用很大的时间梯度进行限制，例如使用边缘截止滤波器。但是，这种操作会造成更大的时间梯度上出现少量闪烁。也就是说，像素级的时域滤波天然存在很多问题与缺陷。往往解决了一个问题之后，都会产生新的问题影响视频的质量。因此，对像素级的时域滤波的一种直接的拓展，就是使用动作补偿时域滤波器，这种滤波器可以有效的过滤出动作在时间线上的轨迹。如果动作的预估是正确的，那么这样的滤波器就能够在时域滤波器的平稳阶段以及边缘都能够准确过滤出动作，并且防止动作模糊现象的产生。但是，在一些情况下，动作预测可能会出现错误，这种情况下，这种滤波方式就会产生失真。例如在产生了镜头突然切换的两帧之间，就会出现动作预测的错误。动作补偿滤波器的例子包括动作向量的块状匹配 (LEE C.)，以及运用光流预测的滤波 (AYDIN T. O., 2014)。

通过使用上述的这些滤波器，理论上来说，现在使用的图像色调映射算法都可以通过增加这些滤波环节，拓展为视频色调映射算法。这也是大多数现在的视频色调映射算法使用的手段。

除了滤波手段之外，第二种利用时间相关性消除视频色调映射过程中出现的失真现象的思路是将时域处理作为一种色调映射的后置操作。也就是说，在这种想法下，使用何种色调映射算法已经无关紧要了，在色调映射之后，利用连续的帧之间的全部与局部信息等得到时域的相关性，然后处理得到的视频，就可以得到消除了各种失真现象的视频。其中一种控制闪烁现象的方式是计算色调映射后每一帧的均值，然后在时域上对每一帧之间的改变做出限制 (GUTHIER B., 2011)，或者将映射后的序列与原 HDR 序列进行比较 (BOITARD R. B. K., 2012)。这些方式通过对不同的帧进行亮度的不同比例的缩放，达到平衡与消除闪烁的目的。但是，这些方法的局限性在于这些亮度的改变只能作用于整幅图像，所以只能调节全局亮度。如果想要调节局部亮度与局部对比度，那么，就需要使用其他的方法进行调整。

大部分调节局部对比度的后置操作方法都是基于动作域的光流预测 (LANG M., 2012)， (DONG X., 2015)， (BONNEEL N., 2015)。这些后置操作算法的好处在于，我们不用像之前一样去考虑色调映射过程是如何完成的，不用去考虑对于不同的色调曲线，需要对滤波器进行哪些改变或者是否需要增加其他操作。特别是对于一些已经进行了色调映射，但是存在失真现象的视频，由于不知道色调曲线是什么，所以很难去针对性地设计滤波器，但是后置操作对于这种情况就可以很好地适应。而这类方法的缺陷在于，对于局部对比度的呈现以及视频的实际效果来看，要弱于滤波器的方法。

噪声

在前面的内容当中，我们把视频色调映射算法的设计问题基本介绍完了，我们可以通过两大类不同的方法来获取与利用视频的时域特性，然后将这种时域特性与图像色调映射算法结合起来，就可以实现视频的色调映射。但是，前面的算法与介绍都是在理想情况下进行的。在实际的生产与应用当中，我们还会遇到其他很多不同的问题，其中一个非常重要的问题就是如何处理噪声。

在 HDR 视频的获取过程中，HDR 摄影设备会产生很多噪声，特别是在阴暗的环境下，产生的噪声更加多。这是由于为了获得更高的动态范围，摄像的曝光时间需要被严格限制。因此，在阴暗的环境下，想要在有限的光照条件下拍摄与捕捉场景中的阴影与

黑暗的区域，就需要将摄影设备传感器的灵敏度调节得非常高。由于传感器灵敏度很高，这就会产生很强的传感器噪声。这些噪声在 HDR 视频当中基本是不可见的，所以并不会影响 HDR 视频的质量与观看体验。但是，经过了色调映射之后，这些噪声在 LDR 视频中会被大幅放大，从而十分影响视频的质量以及观看体验。噪声会在色调映射过程中被放大的原因在于，色调映射也是一种非线性的映射，它会将明亮区域的像素点的亮度降低，然后提升阴暗区域像素点的亮度。也就是说，在色调映射过程中，暗区的噪声被放大了。

为了缓解这些视频噪声，需要对视频进行相应的降噪处理。上一节提到的滤波器方法，可以在执行过程中增加部分操作与设计，这样就可以在色调映射的滤波过程中降低部分噪声 (BENNETT E. P., 2005)， (H., 2006)。但是，这种方法的问题是会额外增加很多的运算，并且并不能够在不使得画面失真的情况下完全去除噪声。并且，如果算法中使用的色调曲线对暗区的增益非常高，那么就会使得没有被去除干净的噪声，以及降噪过程中产生的失真现象被进一步放大。

为了使得噪声不会在色调曲线的映射过程当中显现，一种合理的做法是在了解了噪声特征的情况下，可以对色调曲线的形状进行严格的控制 (EILERTSEN G. M. R., 2015)，(LI J., 2016)。这种情况下，噪声虽然没有被消除，但是却隐藏在了暗区当中，不会被人眼发现，也就不会影响到人们的观看体验。这种方法可以与前面提到的降噪方法同时使用，这样就可以进一步的提高图像的质量与观看效果。

6.3.2 算法综合介绍

视频色调映射算法的不同总体上体现在四个方面，包括：色调曲线，处理方式，算法目标以及时域滤波。同时，由于视频色调映射所需的运算量非常庞大，因此，运算速度也是一个重要的评价指标。

1. 色调曲线：这是色调映射算法的核心。大部分色调映射算法，以及后面我们将要介绍的算法，主要使用以下几种函数作为色调曲线：线性函数，指数/对数函数，sigmoid/s-shaped 函数以及基于灰度图的色调曲线。同时，还要少量色调映射算法在梯度域进行映射。
2. 处理方式：处理方式主要分为两种。一种是全局映射，即对所有的像素使用同样的色调曲线进行映射。第二种是局部映射，在不同的空间位置，使用不同的色调曲线进行映射。
3. 算法目标：色调映射的目的主要分为三类：visual system simulators (VSS), best subjective quality (BSQ) operators and scene reproduction (SRP) operators。(EILERTSEN G. W. R., 2013)， (DUFAUX F., 2016)
4. 时域滤波：时域滤波通常分为两类。第一类是在时域上对色调映射参数进行整体的滤波。第二类是在局部上对每个像素点进行时域滤波。
5. 速度：视频色调映射算法的速度大致分为三个档次：离线，交互式，实时。划分的依据是算法的处理能力以及运算复杂度。实时的算法，可以提供超过 20 帧/秒的运算速度。交互式算法相对弱一些，每秒可以处理几帧。离线算法则速度更慢。由于算法的执行速度与图像的分辨率以及计算机的硬件水平也息息相关，因此这种划分方式也只是一个大概的划分，仅供参考。

6.3.3 代表性算法简介

上一节中，我们对于视频色调映射算法的总体思路以及主要组成部分做了分别的介绍。本节，我们就要将注意力转移到具体的视频色调映射算法上，我们将会介绍一些具体的视频色调映射算法的思路以及实现方法以及主要特点。

1. 视觉适应算法（Visual adaptation TMO）

(FERWERDA J. A., 1996)这是第一个使用时域适应机制的视频色调映射算法。视觉适应算法利用了一系列视觉领域的心理与生理学研究结论建立了一个视觉相应模型。算法的核心方法是：threshold-versus-intensity 方程，这个方程之前大量的实验结果与数据，可以计算出人类视觉范围内的颜色变化，视觉敏感度（能够发现空间上的细节）变化以及时域适应性（时域上明暗变化）变化的阈值。

视觉适应算法的核心思想是，通过在背景亮度下预测出人眼能够感知的颜色，场景清晰度变化阈值，然后只对人眼能够感知到的变化部分，进行色调映射，从而在一定程度上减小了算法的运算量。同时，算法会对视觉敏感度以及时域适应性变化进行额外的处理。

2. 时域适应算法（Time adaptation TMO）

(PATTANAIK S. N., 2000)这种算法充分利用了人类视觉模型的时间适应机制。通过在适应模型后增加一个表现模型，就可以模拟出一幅图像的响应。随后再使用逆表现模型与逆适应模型，就可以将图像显示在 LDR 显示设备上。

在适应模型中，图像中视锥细胞与视杆细胞的静态响应通过 sigmoid 函数被分别计算出，而时域适应性则通过指数平滑滤波器进行建模。并且算法加入了染色与漂白因子，用来对时域上的特征进行着色与漂白。

3. 时域曝光算法（Temporal exposure TMO）

(KANG S. B., 2003)，这种算法首先提出了一种新的获取 HDR 视频的方式：在帧与帧之间，交替的改变曝光时间，从而增大视频的亮度范围。之后，为了创建出 HDR 视频，每一帧利用光流信息进行内插，从而为每一帧提供了两种不同的曝光度，实现了动态范围的提升。并且，针对这类 HDR 图像，文章中也提供了一种算法来对其进行色调映射，从而显示在 LDR 显示设备上。

这种视频色调映射算法基于一种经典的图像色调映射算法：Photographic TMO (REINHARD E., 2002)。通过使用这种算法，再进行时域滤波，从而实现视频色调映射。Photographic TMO 利用了 sigmoid 函数并设计了如下形式的色调曲线：

$$\mathrm{T} = \frac{L_s}{1+L_s}\left(1 + \frac{L_s}{L_{white}^2}\right) \tag{31}$$

其中$L_s = \mathrm{a}\,L/k$，k 表示几何平均数。由于帧与帧之间的平均数会剧烈的改变，从而造成闪烁现象。因此，为了解决闪烁现象，k 会通过固定数量的帧得到，从而缓解闪

烁现象，并且可以在时域上实现有效的统一滤波。

$$\mathrm{k} = \exp(\frac{1}{N}\sum_{s,t}\log(\varepsilon + L)) \tag{32}$$

这里的 s 表示每幅图像中的每一个像素点，t 表示一组连续的帧。ε是一个很小的常数，用来避免奇异值。同时，由于 photographic TMO 的计算成本很大，因此，Goodnight 为这种算法设计了相关的 GPU 加速算法 (GOODNIGHT N., 2003)，使得这种色调映射方法的计算时间大大缩短。

4. 局部适应算法（Local adaptation TMO）

(LEDDA P., 2004)与时域视觉模型类似，局部适应算法同样利用了人眼的时域特性，并且使用了同样的方法，即利用 sigmoid 函数将视觉锥细胞与视觉杆细胞的的响应分开计算。但是，两者的区别是，为了获得局部响应，也就是局部细节，本算法中对每一个像素单独进行这种操作，而时域视觉模型针对每一帧整体进行操作。为了对局部适应性的时域特征进行建模，算法在时域上利用指数滤波器进行建模。并且对于视锥细胞与视杆细胞，明亮区域与阴暗区域，使用了不同的滤波器。

5. 感知效应算法（Perceptual effects TMO）

(KRAWCZYK G., 2005)这种算法同样是建立在 photographic TMO 的基础上。这种算法的主要目的之一，是实现实时的局部色调映射。这种算法利用一些感知效应增强了 photographic TMO 的效果。由于 photographic TMO 需要通过建立高斯金字塔来获取色调映射的局部适应等级，基于这种框架的一些效应处理，只需要增加很少的计算量，就可以实现算法效果的增强。基于这种框架的优势，视觉敏锐度以及光幕照明等算法都被添加进了框架当中。

时域适应性则是通过指数衰减方程（exponential decay function）在时域的适应性等级上进行滤波。具体做法类似于 Interactive walk-through TMO (DURAND F., 2000)。最后，为了提升算法的计算速率以实现实时映射，最终的算法实现使用了 GPU 加速。

6. 梯度域算法（Gradient domain TMO）

(WANG H., 2005)该算法首先也是提供了一种全新的 HDR 视频获取方式。通过一种称之为：分离光圈相机的拍摄设备，这种设备包括 3 个 CCD 传感器相机，可以同时拍摄三种不同亮度范围下同一场景的图像，然后场景的实际亮度可以利用拍摄出的图像通过传统方法重现。而算法的实际目的是通过这种方式获得的 HDR 视频，来研究 HDR 视频在空间域与时域的特性。

对于色调映射算法来说，梯度域算法之前就有所研究。这种算法在不同的位置 s 处，以不同的比例 k，利用比例因子φ_s^k减弱图像的梯度∇H_s^k。

$$\varphi_s^k = \frac{\alpha}{\|\nabla H_s^k\|}\left(\frac{\|\nabla H_s^k\|}{\alpha}\right)^{\beta} \tag{33}$$

改变后的梯度域$G_s = \Phi_s \nabla H_s^k$（Φ 是将所有 k 累加起来的比例因子）之后被用于利用泊松方程来计算色调映射后的 T。

$$\nabla^2 V = \text{div G} \tag{34}$$

与原方法相比，这种算法的差别在于，这里的梯度域是三维的，其中两维表示空间，一维表示时间。解决这种三维泊松问题会使得结果具有时域一致性。但是，对于色调映射问题来说，处理过程中只对空间特性进行梯度减弱，因为如果进行了时域的减弱，就会产生运动模糊。

7. 区域匹配算法 （Block match TMO）

(LEE C., Gradient domain tone mapping of high dynamic range videos, 2007)这种算法的思想与梯度域算法类似，即色调映射在梯度域执行，并且梯度是三维的，两维表示空间，一维表示时间，梯度在计算过程中通过压缩得到减弱。与梯度域算法不同的是，梯度域算法只在空间域的两维上进行梯度减弱，区域匹配算法在时间域的维度上也进行了梯度减弱。为了在实现时间域梯度减弱的过程中不产生运动模糊，算法中又利用区域匹配以实现运动补偿，从而消减运动模糊的影响。

记像素点 s 处，从第 t-1 到第 t 帧的动作的估计向量为$v_{s,\ t}$，对应的代价函数为：

$$\text{C} = \sum_s \left(L_{s,t} - L_{s-v_{s,t},t-1}\right)^2 \tag{35}$$

当上一帧的像素点的值与这一像素点在下一帧预计应该在的位置的像素点值存在差异时，代价函数的值就会增大，也就意味着这个运动估计是不可靠的。代价 C 会被用于泊松方程以平滑动作路径上的点的梯度。这样操作之后，就可以增强相邻帧之间的时域相关性，并减弱噪声。也就是说，这种算法不仅不会像之前的梯度域算法一样产生运送模糊，并且还可以减弱闪烁与时域不相关问题。

8. 视网膜模型算法（Retina model TMO）

(BENOIT A., 2009)这种算法的思想来源与模拟人类视觉系统的处理方式，也就是视网膜的处理方式。在人眼对图像的的处理过程中，眼部细胞首先接受光感受器的响应，然后将刺激传递到神经节细胞，最后由视网膜进行空间域与时域的滤波，进而实现视觉的生成。这种算法模拟了视网膜的底层处理，包括局部适应性。

首先，利用 sigmoid 函数可以获得图像的局部响应，通过空间低通滤波器可以获得图像的局部适应等级。局部响应通过时空滤波器，可以模拟出 Outer Plexiform Layer (OPL) (HÉRAULT J., 2007)的处理方式。OPL 滤波器可以被认为是两个低通时空滤波器的差，并且将频谱进行了白化滤波。也就是说，它将高频部分中的低频信号与低频部分中的高频信号进行了过滤。这种滤波器通过过滤高频部分，从而消减了噪声；并且通过过滤时域的局部适应等级，从而提供了更好的时域稳定性。在通过了时空滤波器之后，响应会通过神经节细胞响应模型，这个模型与算法的第一步很类似，都是通过 sigmoid 函数，区别是使用了不同的参数。

整个算法的最后一步是颜色的重建。在算法的最一开始，颜色信息会在初始化阶段，通过多路处理被提取出来。而在最后一步，颜色信息会被还原到图像当中。整个操作的目的是实现颜色的一致性，防止在算法中间的滤波过程中，对颜色信息产生不可逆的修改。这种思想，来源于视网膜的处理方式 (MEYLAN L., 2007)。

9. 闪烁消除模型（Flicker reduction TMO）

(GUTHIER B., Flicker reduction in tone mapped high dynamic range video, 2011)这种算法实际上并非完整的视频色调映射算法。这种算法的应用对象是已经经过色调映射的视频，模型通过检测与处理，可以将视频中存在的闪烁现象减弱或去除。

在色调映射后的视频序列中，首先检测是否存在闪烁现象。算法的检测方法是通过检测连续帧之间的 k 值之差是否超过了阈值 H，来进行判断。其中，K 值就是时域曝光模型中的 k，而阈值则通过 Steven's Power Law (S., 1962)计算，$\mathrm{H} = \Delta\mathrm{L} = \mathrm{a}L^{1/3}$，a 通过实验决定。

$$\mathrm{k} = \exp(\frac{1}{N}\sum_{s,t}\log(\varepsilon + L)) \tag{36}$$

对于大于阈值的帧，亮度迭代地下降，知道帧的 k 值范围落在合理范围之内$k_t \in \{k_{t-1} - H, k_{t-1} + H\}$。对于只存在全局亮度问题，也就是全局闪烁问题的色调映射后的视频序列，这种算法都是有效的。

10.时域相关模型（Temporal coherence TMO）

(BOITARD R. B. K., 2012)除了时域闪烁之外，视频色调映射中海油两个潜在的需要注意的问题，一个是物体不连续问题，用于描述在色调映射物体的亮度变化问题以及由于不同的帧的统计信息不同，而导致的与原视频序列中物体的变化不一致的问题。第二个问题是亮度不连续，用于描述视频整体的亮度在时域上的改变，例如经过了色调映射后，原视频中亮度最高的一帧可能就不再是最亮的一帧。

为了使色调映射算法能够克服这些问题，Boitard 等人提出了时域相关模型。这种模型与前一个提到的闪烁消除模型一样，也不是完整的色调映射算法，而是用于从色调映射后的视频序列中消除时域问题的一类附加修正算法。对于每一帧 t，首先计算它在色调映射之前的 k 值k_t^L以及色调映射之后的 k 值k_t^T。计算完成后，利用全局系数τ将色调映射后的视频的每一帧的 k 值都固定位原 HDR 视频序列中 k 值的最大值，$\tilde{T} = \tau T$。

$$\mathrm{k} = \exp(\frac{1}{N}\sum_{s,t}\log(\varepsilon + L)) \tag{37}$$

$$\tau = \frac{k_t^L k_{max}^T}{k_{max}^L k_t^T} \tag{38}$$

接着再用一个调制系数，使得τ能够在很小的范围内浮动

$$\tau = \tau_{min} + (1 - \tau_{min})\tau \tag{39}$$

由于这种算法也是一种附加操作，因此，这种算法也使用与所有只存在全局亮度问题的色调映射算法。这种算法利用了整个视频序列的信息来解决亮度问题，并且将每一帧都与原 HDR 视频进行比较。这样的操作会造成视频的整体亮度出现下降，因为为了保持视频亮度的相对不变，视频中的大多数帧的亮度在操作过程中都会下调。

11.混合模型（Hybrid TMO）

(SHAHID H., 2015)混合模型顾名思义，利用了两种不同的色调映射算法对同一帧进

行处理。首先，每一帧会通过 directional filter bank edge-detection algorithm (ANAND S., 2012)进行处理，将图像分为包含剧烈改变的区域以及不包含剧烈改变的区域。包含剧烈改变的区域随后使用 iCAM06 TMO (KUANG J., 2007)算法进行色调映射，这种算法通过局部操作保留了图像的细节。图像的剩余区域通过直方图校正算法 (WARD LARSON G., 1997)进行处理。最后，为了保持视频的时域性质，映射后的帧的亮度会进行比例缩放，进而使得连续的帧之间的强度差值不会超过预先设定的阈值，具体操作类似于闪烁消除模型。

12.噪声感知模型（Noise-aware TMO）

(EILERTSEN G. M. R., Real-time noise aware tone mapping, 2015)这种算法用于处理视频获取过程中出现的噪声。算法的目的是利用一种有效的时域稳定的视频色调映射算法，在加强视频局部对比度的同时，不提升视频噪声，使视频噪声保持不可见。

算法中，利用式子（4）实现局部色调映射，并且算法中还使用了一种特别设计的滤波器。这种滤波器基于标准的用于噪声消除的边缘保留滤波器，它可以在一些情况下找出图像中的失真区域。色调映射中的细节抽取则使用了不同的思路。为了能够抽取出图像的细节并且不引起失真，算法中使用了一种简化版的双边滤波以及各向异性扩散滤波。这种方法可以解决与 0 阶各向异性相关的重构问题。

对于色调映射来说，色调曲线用于将原始图像序列的全局对比度尽可能地保存，因此在设计色调曲线的时候，实际上是在解决一个最优化问题，优化的目标是减少由于亮度映射带来的对比度的畸变。对于输入的对比度 G 以及映射后的对比度$\tilde{G}$，畸变的平方的期望$\left(G-\tilde{G}\right)^2$可以利用对比度相对于对数域的亮度的分布$p(G|L)$进行表示：

$$\mathrm{E}\left[\left\|G-\tilde{G}\right\|_2^2\right]=\int p(L)\int\left(G-\tilde{G}\right)^2 p(G|L)dGdL \tag{40}$$

上式提供了一个最优化问题，可以通过将色调曲线参数化表示以及使用图像直方图来表示 P(L)来很快地解决。整体的概念与显示适应模型 (MANTIUK R., 2008)中的对比度优化过程十分类似，但是本算法中使用的方式减少了很多计算复杂度。为了保证时域的平滑改变，色调曲线的节点会在时域上利用低通 IIR 或者边缘截止滤波器进行滤波。

由于色调曲线的映射会将原视频中不可见的噪声变得可见，为了防止这一现象的发生，算法中使用了噪声感知技术。这一技术的想法是在色调映射的优化模型中，也就是式（22）的优化模型中，引入表示图像噪声的项，在最优化的过程中，这一项就会被不断削弱。同时，算法中还将提取出的细节层进行了缩放，以减弱可见噪声。基于这两种方法，算法能够将可见的噪声减弱至人眼无法观察到的程度，进而提升了图像的质量。

参考文献：

ANAND S., T. T. (2012). Edge detection using directional filter bank. *International Journal of Applied Information Systems 1*, pp. 21-27.

AYDIN T. O., S. N. (2014). Temporally coherent local tone mapping of HDR video. *ACM Trans. Graphics 33, 6*, pp. 1–13.

BENNETT E. P., M. L. (2005). Video enhancement usingper-pixel virtual exposures. *ACM Trans. Graphics 24, 3*, pp. 845–852.

BENOIT A., A. D. (2009). Spatio-temporal tone mapping operator based on a retina model. *Computational Color Imaging Workshop(CCIW'09)*, (pp. 12–22).

BOITARD R., B. K. (2012). Temporal coherency for video tone mapping. *SPIE 8499, Applications of Digital Image Processing XXXV*, (pp. 84990D–84990D–10). San Diego.

BOITARD R., C. R. (2014). Survey of temporal brightness artifacts in video tone mapping. *HDRi2014-Second International Conference and SME Workshop on HDR imaging*, (pp. 9-15).

BONNEEL N., T. J. (2015, Oct.). Blind video temporal consistency. *ACM Trans. Graph. 34, 6*, pp. 196:1–196:9.

DONG X., B. B. (2015). Region-based temporally consistent video post-processing. *The IEEE Conference on Computer Vision and Pattern Recognition(CVPR).*

DUFAUX F., C. P. (2016). *High Dynamic Range Video: From Acquisition, to Display and Applications, vol. 1.* Academic Press.

DURAND F., D. J. (2000). Interactive Tone Mapping. In *Rendering Techniques 2000: Proceedings of the Eurographics Workshop* (pp. 219–230). Brno, Czech Republic.

EILERTSEN G., M. R. (2015, Oct). Real-time noise aware tone mapping. *ACM Trans. Graph. 34, 6*, pp. 198:1–198:15.

EILERTSEN G., W. R. (2013). Evaluation of Tone Mapping Operators for HDR-Video. *Computer Graphics Forum 32, 7*, pp. 275–284.

EILERTSEN G.R. K., UNGER J.MANTIUK. (2015). Real-time noise-aware tone mapping. ACM Trans. Graph. 34, 6.

EILERTSEN G.R., MANTIUK R. K., UNGER J.WANAT. Evaluation of Tone Mapping Operators for HDR-Video. Computer Graphics Forum 32, 7 (2013).

FERWERDA J. A., P. S. (1996). A model of visual adaptation for realistic image synthesis. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (pp. 249–258.). New York, NY, USA,: SIGGRAPH '96， ACM.

GOODNIGHT N., W. R. (2003). Interactive time-dependent tone mapping using programmable graphics hardware. *Proceedings of the 14th Eurographics Workshop on Rendering*, (pp. 26-37). Aire-la-Ville, Switzerland, Switzerland.

GUTHIER B.S., EBLE M., EFFELSBERG W.KOPF. (2011). Flicker reduction in tone mapped high dynamic range video. Proc. SPIE.

H., V. H. (2006). Encoding of high dynamic range video with a model of human cones. *ACM Trans. Graphics 25*, pp. 1380-1399.

HÉRAULT J., D. B. (2007). *Modeling Visual Perception for Image Processing.* Berlin, Heidelberg: Springer Berlin Heidelberg.

KANG S. B., U. M. (2003). High dynamic range video. *ACM Trans. Graph. 22, 3*, pp. 319–325.

KRAWCZYK G., M. K.-P. (2005). Perceptual effects in real-time tone-mapping. *Proceedings of the 21st Spring Conference on Computer Graphics*, (pp. 635–645).

KUANG J., J. G. (2007). icam06: A refined image appearance model for {HDR} image rendering. *Journal of Visual Communication and Image Representation 18, 5* , pp. 406-414.

LANG M., W. O. (2012, July). Practical temporal consistency for image-based graphics applications. *ACM Trans. Graph. 31, 4*, pp. 34:1–34:8.

LEDDA P., S. L. (2004). A local model of eye adaptation for high dynamic range images. *International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa 3*, (pp. 151-160).

LEE C., K. C.-S. (2007). Gradient domain tone mapping of high dynamic range videos. *International Conference on Image Processing.*

LI J., S. O. (2016). Novel real-time tone-mapping operator for noisy logarithmic cmos image sensors. *Journal of Imaging Science and Technology 60, 2*, pp. 1–13.

MANTIUK R., D. S. (2008). Display adaptive tone mapping. *ACM Trans. Graphics 27, 3*, pp. 68:1–68:10.

MEYLAN L., A. D. (2007, Sep). Model of retinal local adaptation for the tone mapping of color filter array images. *J. Opt. Soc. Am. A 24, 9*, pp. 2807–2816.

PATTANAIK S. N., T. J. (2000). Time-dependent visual adaptation for fast realistic image display. *Proc. SIGGRAPH 00 (2000), Annual Conference Series*, (pp. 47–54).

REINHARD E., S. M. (2002, 7). Photographic tone reproduction for digital images. *ACM Trans. Graph. 21,3*, pp. 267–276.

S., S. (1962). The surprising simplicity of sensory metrics. *American psychologist*, p. 29.

SCHLICK, C. (n.d.). Quantization techniques for visualization of high dynamic range pictures. *Springer-Verlag*, pp. 7-20.

SHAHID H., L. D. (2015, 1). A new hybrid tone mapping scheme for high dynamic range (hdr) videos. *IEEE International Conference on Consumer Electronics (ICCE)*, (pp. 351–352).

WANG H., R. R. (2005). High dynamic range video using split aperture camera. *IEEE 6th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras.* Washington, DC, USA.

WARD LARSON G., R. H. (1997). A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Trans. Visualization and Computer Graphics 3, 4*, pp. 291-306.